An Introduction to Progressive Delivery

James Governor @monkchips co-founder

RedMonk


STA

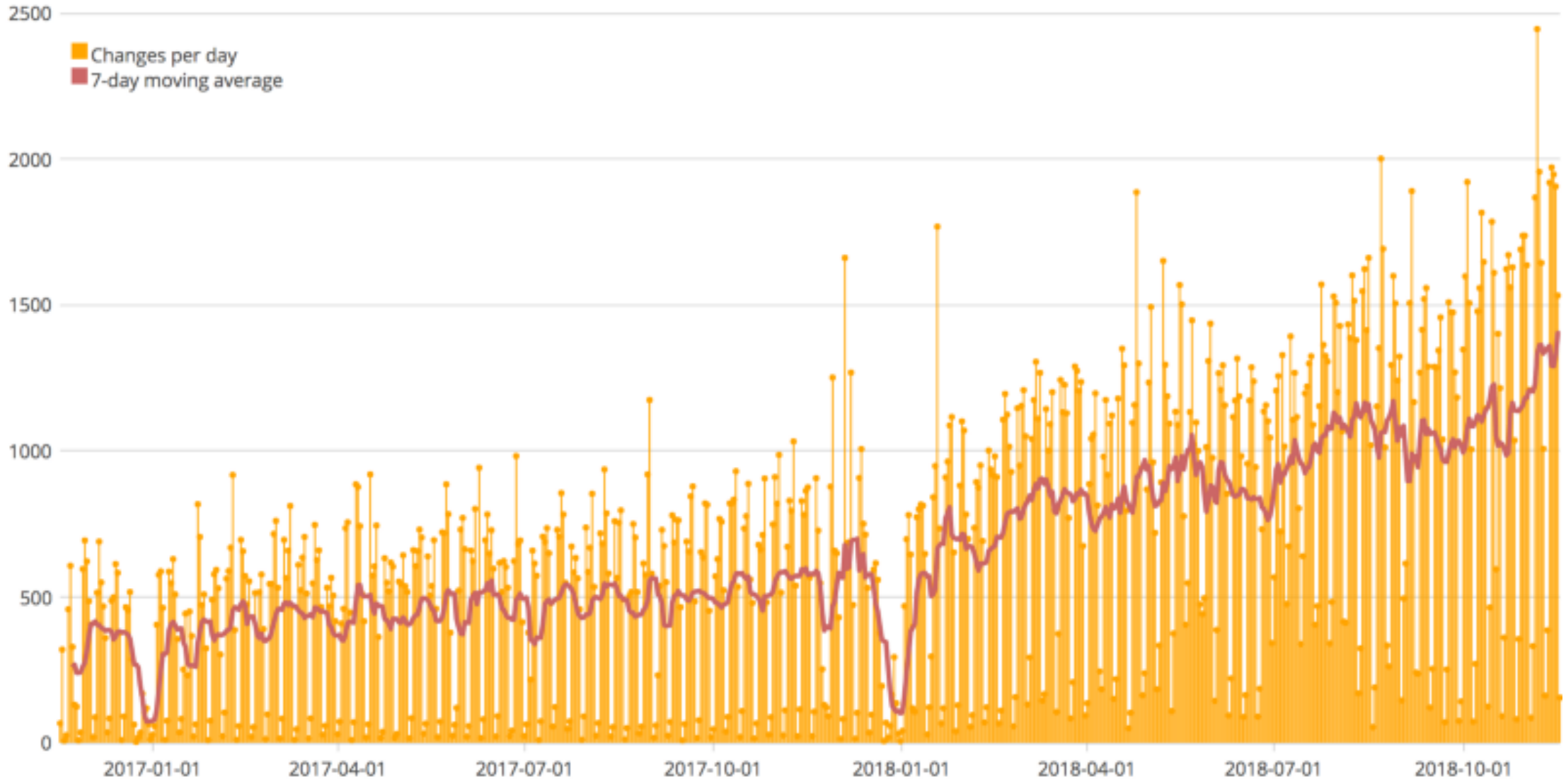
what even is Istio for?

What is Progressive Delivery?

Technology underpinnings

Early adopters

@monkchips


2500
2000
1500
1000
500
0
Changes per day
7-day moving average
2017-01-01
2017-04-01
2017-07-01
2017-10-01
2018-01-01
2018-04-01
2018-07-01
2018-10-01


DORA
DEVOPS RESEARCH & ASSESSMENT

DevOps

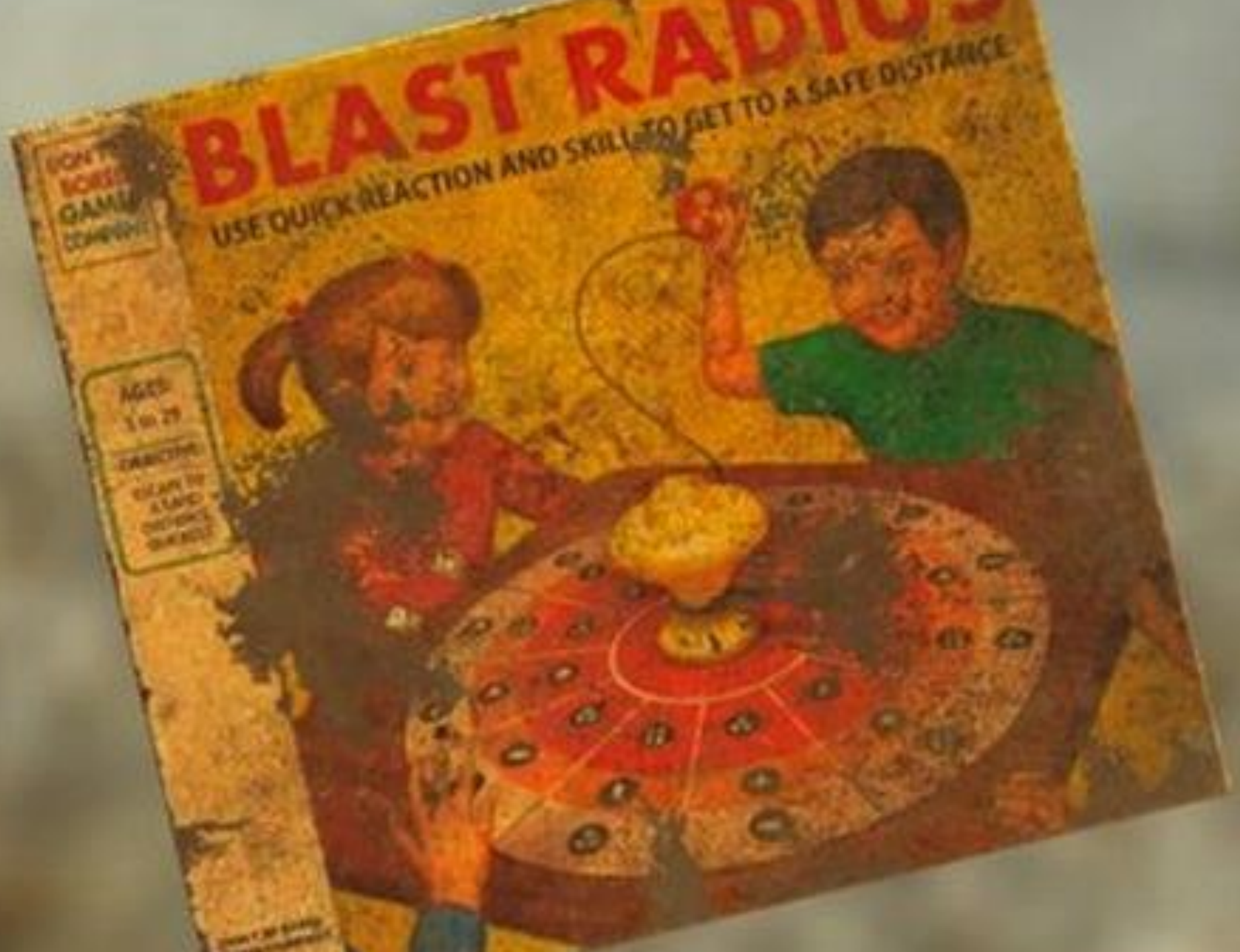
CI/CD


ORBO


Anfim
Standl
20
DA WO'S AN
KAFFEE GIBT
äthiopien
Koke
HONEY


COMPAK
Standl
20
DA WO'S AN
KAFFEE GIBT
Costa
Rica

Microsoft Azure


BLAST RADIUS
USE QUICK REACTION AND SKILL TO GET TO A SAFE DISTANCE


T
SPI

“Progressive Delivery"

What is Progressive Delivery?

Progressive Delivery is Continuous Delivery with fine-grained control over the blast radius.

Building blocks

- User segmentation
- Traffic management
- Observability
- Automation

https://redmonk.com/jgovernor/2018/08/06/towards-progressive-delivery/

"Progressive Delivery is the next step after Continuous Delivery, where new versions are deployed to a subset of users and are evaluated in terms of correctness and performance before rolling them to the totality of the users and rolled back if not matching some key metrics."

- Carlos Sanchez, CloudBees

deploy!=release

COMCAST


Coca-Cola
mentos
The second before everything ceased to exist


aws

Regional isolation

Not too long after we launched Availability Zones, we also launched our second Region, EU (Ireland). Early in the design, we considered operating a seamless global network, with open connectivity between instances in each Region. Services such as S3 would have behaved as "one big S3," with keys and data accessible and mutable from either location.

The more we thought through this design, the more we realized that there would be risks of issues and errors spreading between Regions, potentially resulting in large-scale interruptions that would defeat our most important goals:

- To provide the highest levels of availability
- To allow Regions to act as standby sites for each other
- To provide geographic diversity and lower latencies to end users

Our experience with the benefits of Availability Zones meant that instead we doubled down on compartmentalization, and decided to isolate Regions from each other with our hardest boundaries. Since then, and still today, our services operate autonomously in each Region, full stacks of S3, DynamoDB, Amazon RDS, and everything else.

https://www.allthingsdistributed.com/2018/03/ten-years-of-aws-compartimentalization.html

We roll out a new service to 5% of our customers first.

What sort of users choose to use this feature? We roll out the service then leverage our logs to understand the behaviours of the system and users. Logs are integral to understanding how new code is being shipped, how you do A/B testing in production. We do testing in production."

Bruno Kurtic, founding vp

sumo logic

CLOUDFLARE®

Istio for Canaries. Jason Yee https://www.youtube.com/watch?v=MGLDRKiXNf0

Think about **which** users you're rolling out a service to, when, in which order and why.

Do Japanese customers use services differently?

Grubhub canary deploys to small cities first

SRE Book Golden signals – latency, errors, traffic, saturation

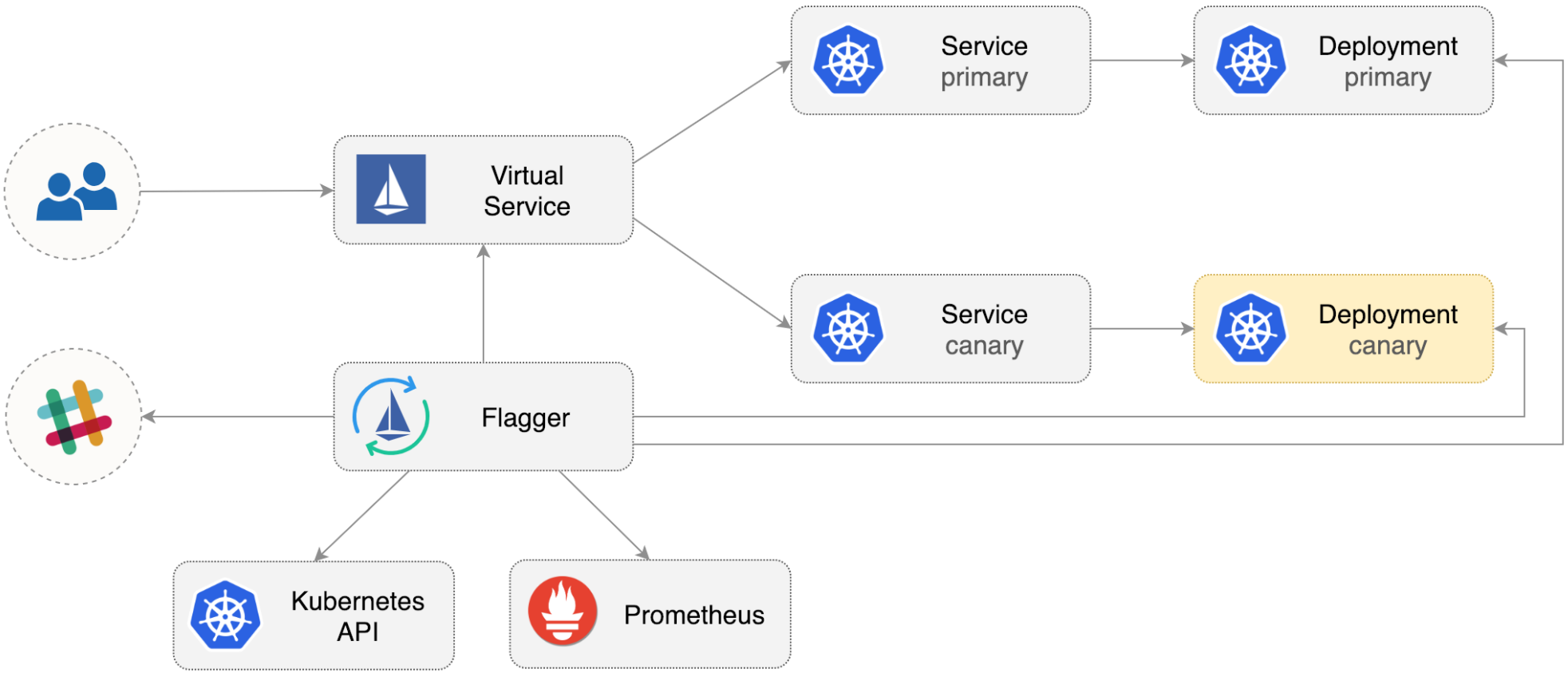

developer
experience

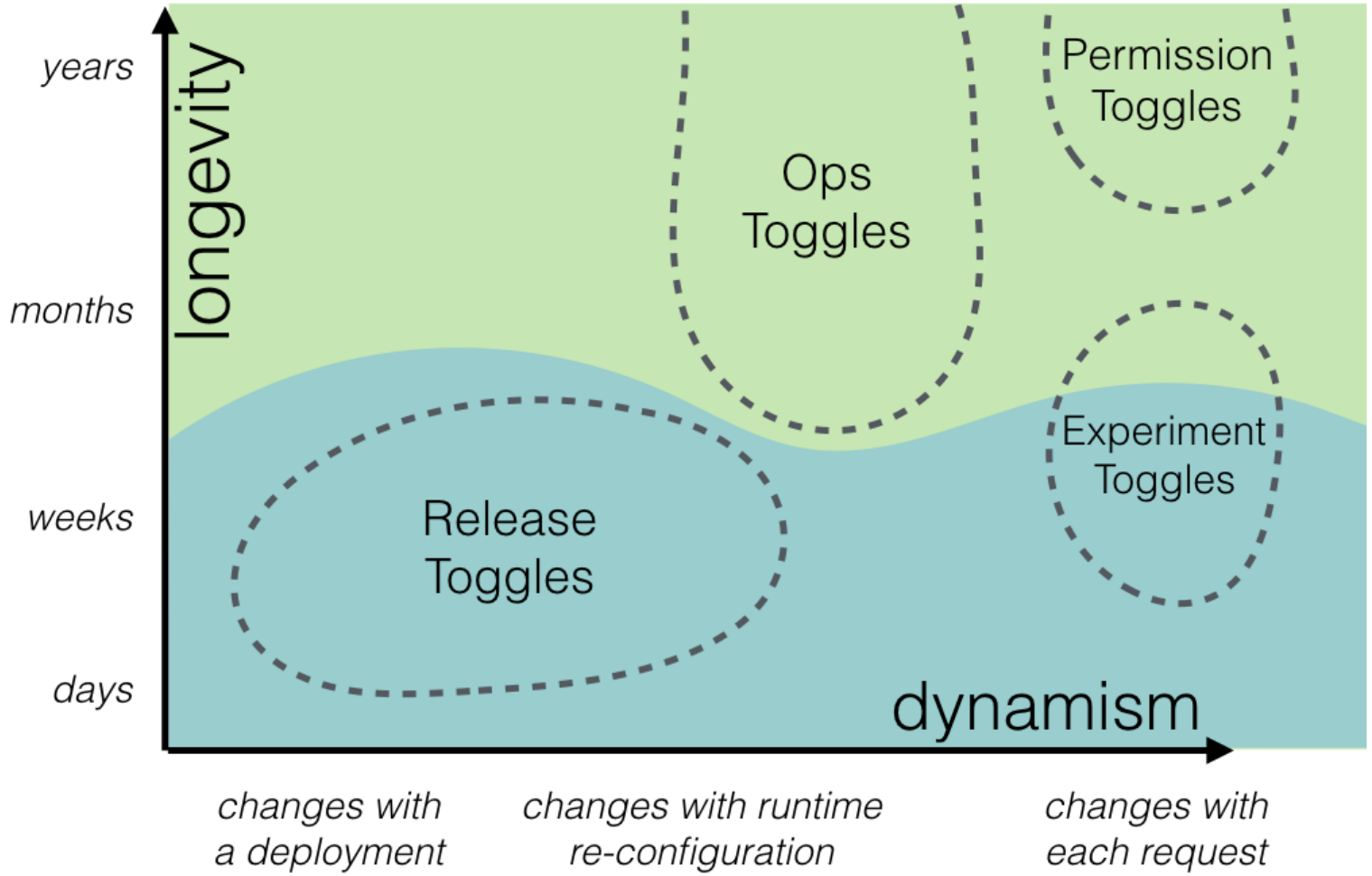
Clip slide

Introducing Flagger

Flagger is a Kubernetes operator that automates the promotion of **canary deployments** using Istio routing for **traffic shifting** and Prometheus **metrics** for canary analysis.

Flagger implements a **control loop** that gradually shifts traffic to the canary while measuring key performance indicators. Based on the KPIs **analysis** a canary is promoted or aborted.


Service
primary
Deployment
primary
Virtual
Service
Service
canary
Deployment
canary
Flagger
Kubernetes
API
Prometheus



Pete Hodgson at martinfowler.com

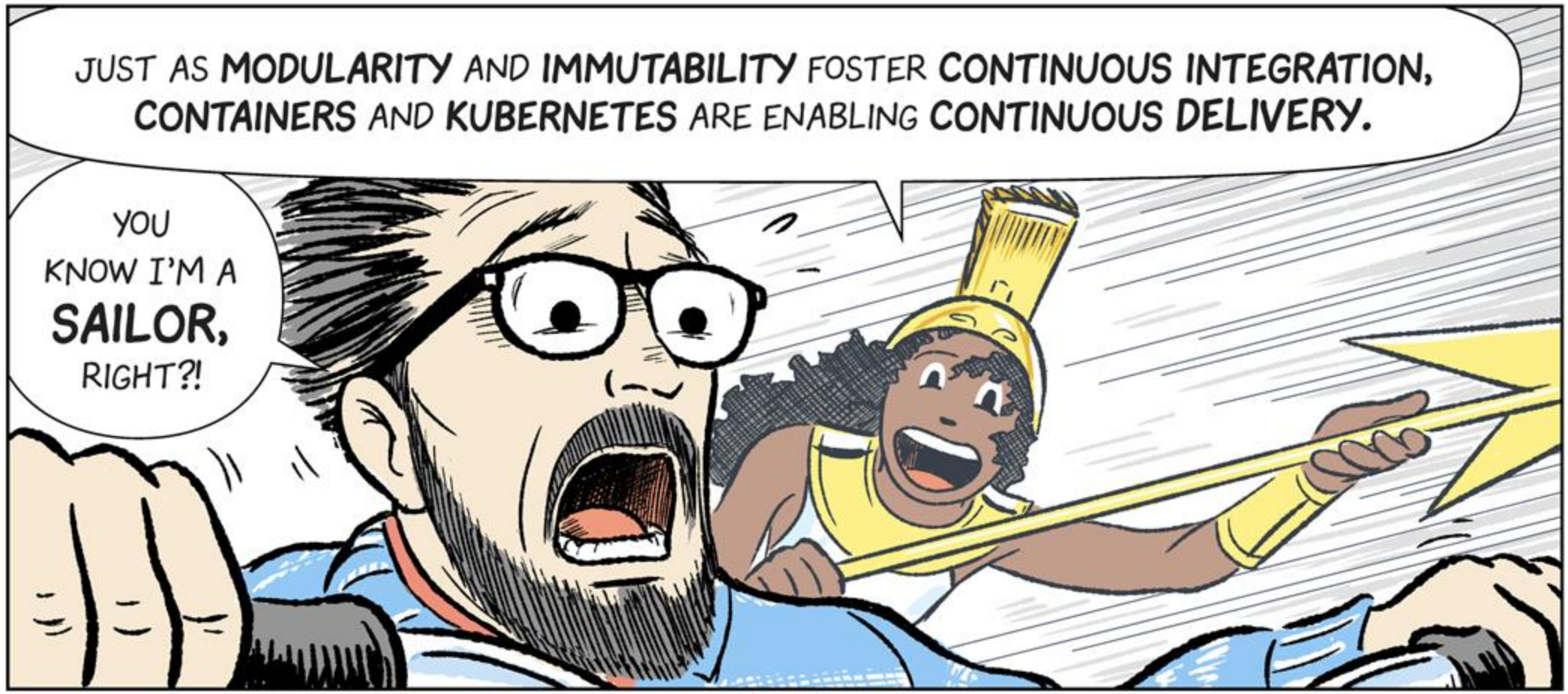
LaunchDarkly

If statement for features

Release Progression – progressively increasing the number of users that are able to see (and are impacted by) new features.

Delegation – progressively delegating the control of the feature to the owner that is most closely responsible for the outcome.

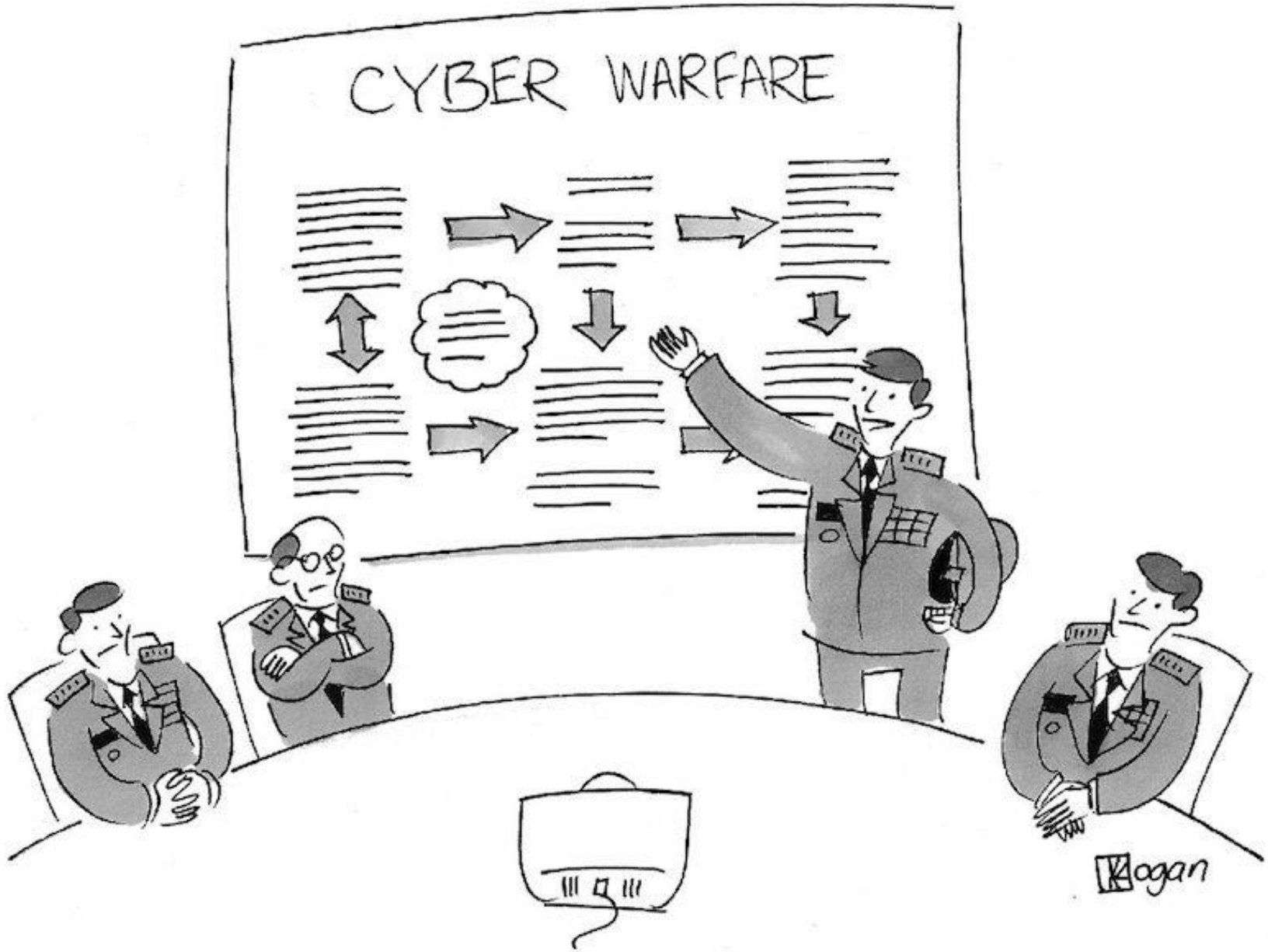
Google Cloud


JUST AS MODULARITY AND IMMUTABILITY FOSTER CONTINUOUS INTEGRATION, CONTAINERS AND KUBERNETES ARE ENABLING CONTINUOUS DELIVERY.
YOU KNOW I'M A SAILOR, RIGHT?!


But the end result is the same, is it not?


TARGET

IBM **Cloud**



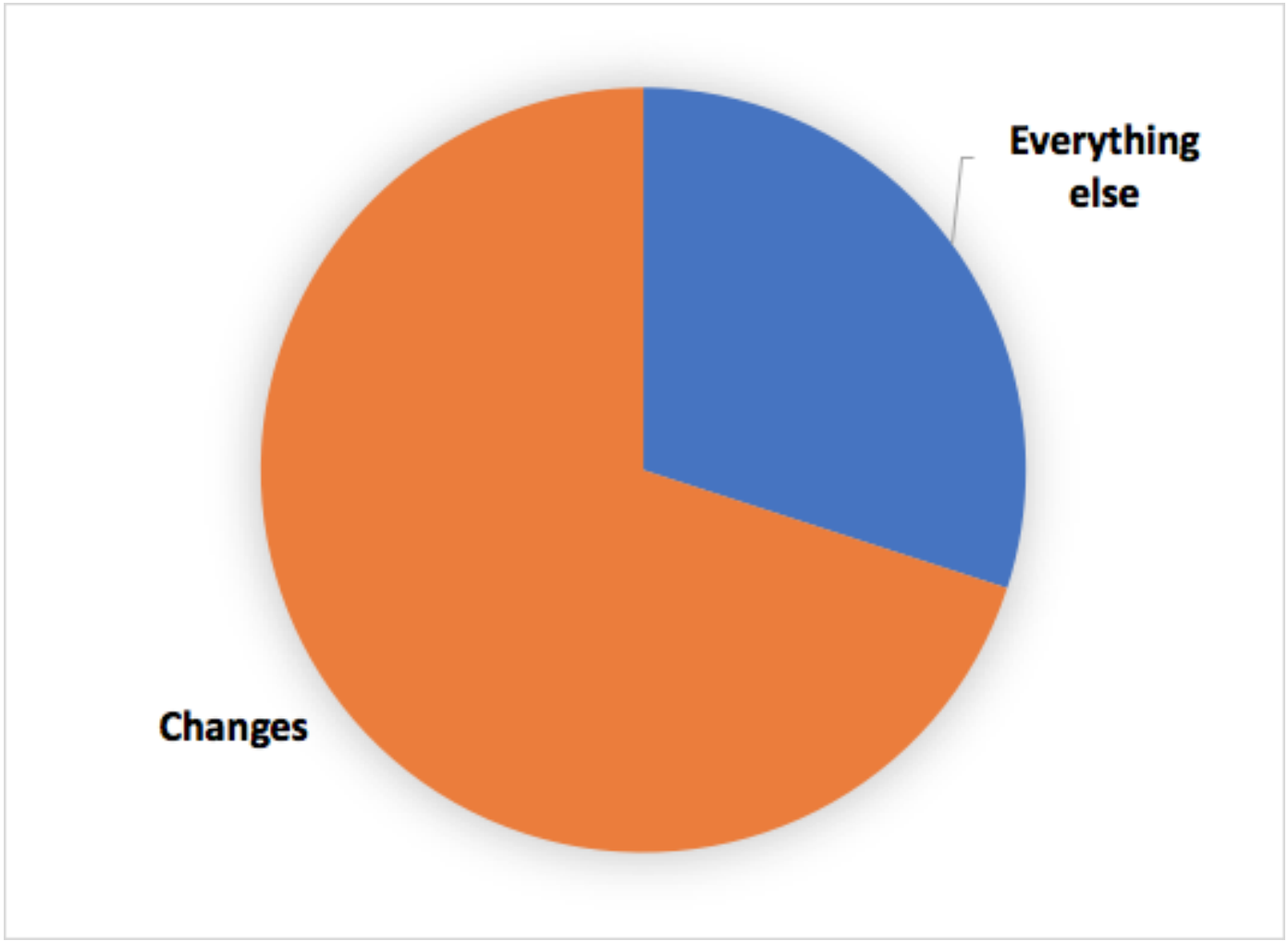
"First, we inundate them with quarterly Kubernetes releases."

Service meshes give you

Advanced service routing and traffic shifting
Easier Rollbacks
Automatic metrics, logs and traces (Prometheus)

And so, Progressive Delivery

Outages at Expedia



Subbu Alamaraju https://m.subbu.org/incidents-trends-from-the-trenches-e2f8497d52ed

Subbu Allamaraju @sallamar · Feb 27

Replying to @monkchips @qconlondon

Progressive delivery is a compartmentalization strategy

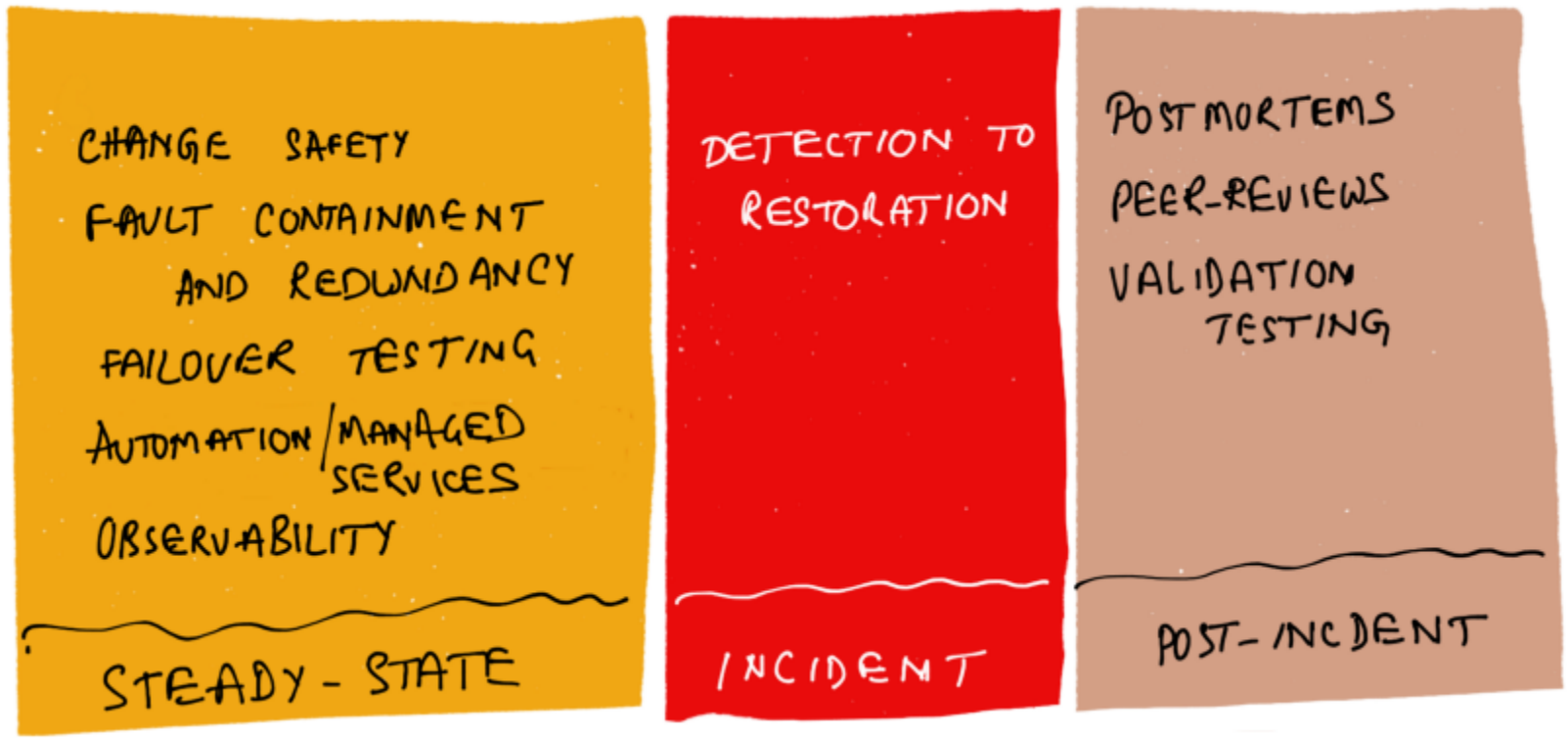
1 1

Subbu Allamaraju @sallamar · Feb 27

Replying to @monkchips @qconlondon

Yes the idea is to compartmentalize change first, then rollback, and then failover (this is a **progressive** journey)

1


CHANGE SAFETY
FAULT CONTAINMENT AND REDUNDANCY
FAILOVER TESTING
AUTOMATION / MANAGED SERVICES
OBSERVABILITY
STEADY-STATE
DETECTION TO RESTORATION
INCIDENT
POSTMORTEMS
PEER-REVIEWS
VALIDATION TESTING
POST-INCIDENT

GitOps

Olly

align User Experience and Developer Experience

enhanced A/B testing

experimentation and feature flags

manage deployment target complexity

reduce business risk

decouple deployment from service activation

Bring the business back into control of rollouts – towards product management focus

Turn pace of software delivery into a service management feature

Avoid rollbacks and emergency fixes across mass user populations

Debug in production, across a limited user population

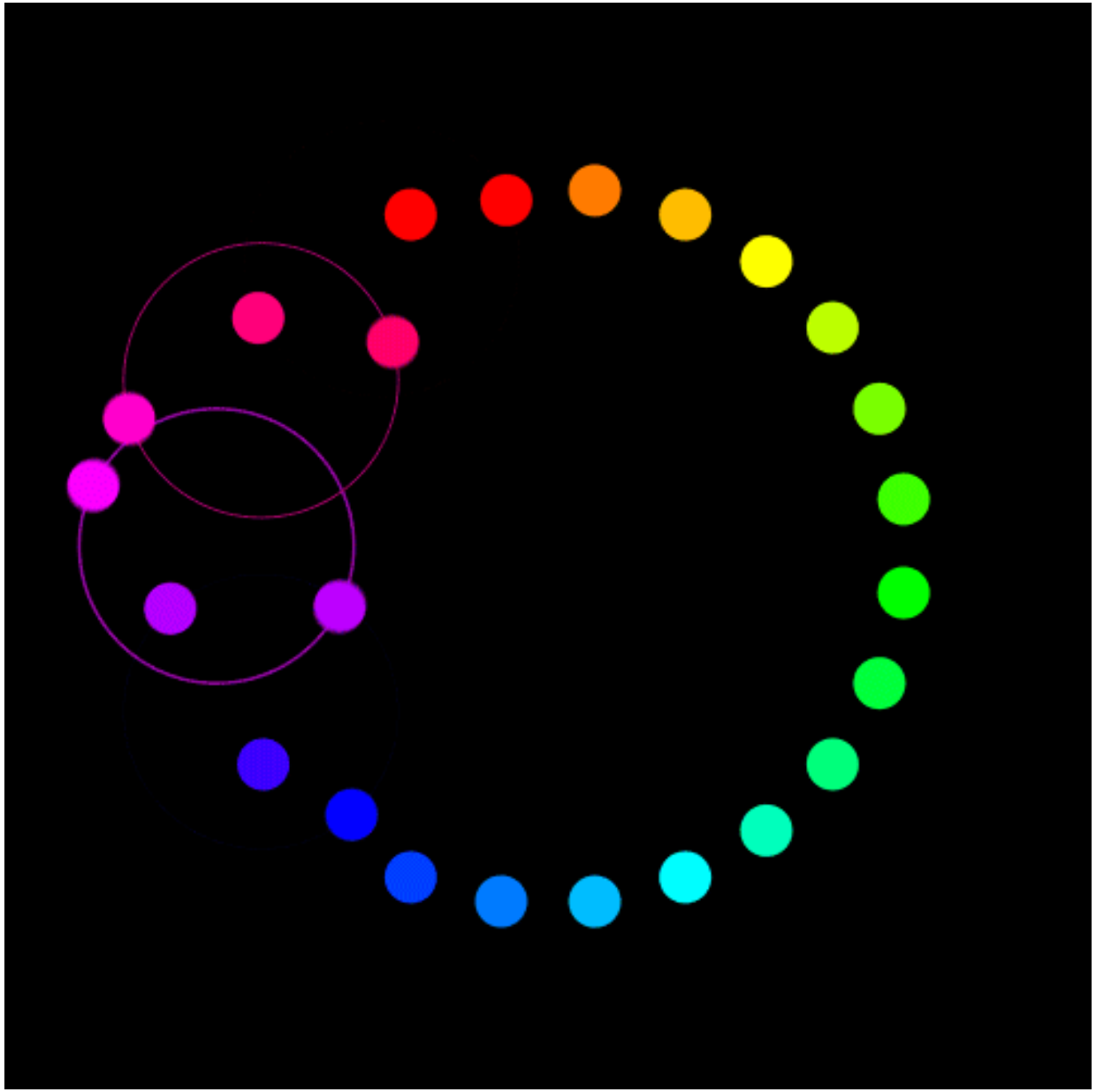
Deploy!=Release

Think about **which** users you're rolling out a service to, when, in which order, and why.

Finally a reason to consider Istio

Related – GitOps, Observability

Use the abundance!



Additional Reading:

Towards Progressive Delivery

Progressive Delivery at Sumo Logic

New Kingmakers, How Developers Conquered The World by Stephen O'Grady, RedMonk – free ebook.

Observability: Charity Majors (@mipsytipsy), Cindy Sridharan (@copyconstruct), Jaana B. Dogan (@rakyll)

GitOps - Alexis Richardson, (@monadic) founder of weave.works. What You Need to Know